GUARDS - Innovative Application of Game Theory for National Airport Security

James Pita, Milind Tambe, Christopher Kiekintveld*, Shane Cullen, Erin Steigerwald*****
University of Southern California, Los Angeles, CA 90089
*University of Texas at El Paso, El Paso, TX 79968
**APEX STORE-Technology Lead: Science and Technology Directorate,
Department of Homeland Security
***Program Manager: Transportation Security Administration

Abstract

We describe an innovative application of a novel game-theoretic approach for a *national scale* security deployment. Working with the United States Transportation Security Administration (TSA), we have developed a new application called GUARDS to allocate the TSA's limited resources across hundreds of security activities to provide protection at over 400 United States airports. Similar security applications (e.g., ARMOR and IRIS) have focused on one-off tailored applications and one security activity (e.g. checkpoints) per application, GUARDS on the other hand faces three new key issues: (i) reasoning about hundreds of heterogeneous security activities; (ii) reasoning over diverse potential threats; (iii) developing a system designed for hundreds of end-users. Since a national deployment precludes tailoring to specific airports, our key ideas are: (i) creating a new game-theoretic framework that allows for heterogeneous defender activities and compact modeling of a large number of threats; (ii) developing an efficient solution technique based on general purpose Stackelberg game solvers; (iii) taking a partially centralized approach for knowledge acquisition. The scheduling assistant has been delivered to the TSA and is currently undergoing evaluation for scheduling practices at an undisclosed airport. If successful, the TSA intends to incorporate the system into their unpredictable scheduling practices nationwide.

1 Introduction

The United States Transportation Security Administration (TSA) is tasked with protecting the nation's transportation systems, including the over 400 airports [TSA, 2011]. These airports serve approximately 28,000 commercial flights per day and approximately 87,000 total flights [AIR, 2011]. To protect this large transportation network, the TSA employs approximately 48,000 Transportation Security Officers [TSA, 2011]. These Security Officers are responsible for implementing security activities at each individual airport.

While many people are aware of common security activities, such as individual passenger screening, this is just one of many security layers TSA personnel implement to help prevent potential threats [TSA, 2011]. These layers can involve hundreds of heterogeneous security activities executed by limited TSA personnel leading to a complex resource allocation challenge. Unfortunately, TSA cannot possibly run every security activity all the time and thus must decide how to best allocate its resources among the layers of security.

To aid the TSA in scheduling resources in a risk-based manner, we developed a software system, Game-theoretic Unpredictable and Randomly Deployed Security (GUARDS), in collaboration with TSA subject matter experts. GUARDS utilizes a Stackelberg game framework, which has previously been shown to be an advantageous model for security domains [Jain *et al.*, 2010b], where one agent (the leader) must commit to some strategy first and a second agent (the follower) can make his decision with knowledge of this commitment. Here, the TSA acts as a defender (i.e. the leader) who has a set of security activities to protect a set of targets and a limited number of resources to assign to these activities. This approach then models a motivated attacker's ability to observe the TSA's resource allocations before attempting to attack an airport target. The goal of our analysis is to compute the optimal mixed strategy for the TSA to commit to in order to provide them with a risk-based, randomized schedule for allocating their limited resources.

The fundamental novelty in GUARDS, compared to previous applications (e.g., IRIS) [Jain *et al.*, 2010b] of such game-theoretic approaches, is the potential national scale deployment. Given that previous approaches dealt with a single standalone location, this national deployment raises three key issues: (i) appropriately modeling the game; (ii) efficiently solving the game; (iii) acquiring knowledge for the game.

From the modeling perspective, traditional models of security games [Yin *et al.*, 2010] are no longer appropriate models. In fact, the TSA's domain has the following additional features beyond traditional security games: (i) heterogeneous security activities for each target; (ii) heterogeneous threats for each target. To appropriately model these additional features we created a novel game-theoretic model, which is referred to as "Security Circumvention Games" (SCGs), and cast the TSA's challenges within this model. In the creation of SCGs we provide the following contributions beyond traditional security games: (i) the ability for defenders to guard targets with more than one type of security activity (heteroge-

neous activities); (ii) the ability for attackers to choose threats designed to circumvent specific security activities.

Given our new model, it was necessary to design an efficient solution technique since previous solution techniques [Jain *et al.*, 2010a] for security games are no longer directly applicable. Thus, we developed an efficient solution technique where we use a general Stackelberg game solver known as DOBSS [Jain *et al.*, 2010b] and rely on a compact representation of SCGs. This is opposed to the use of a tailored domain specific Stackelberg game solver that may not be applicable to all airports nationwide.

For knowledge acquisition, in consideration of national deployment for the TSA, we face two unique constraints. First, headquarters cannot do centralized planning where they create a single optimal mixed strategy (security policy) that will be applicable to all airports. Second, TSA wants to maintain a common standard of security among airports. This precludes an entirely decentralized approach where each airport is completely in charge of creating their security policy. We took a partially centralized approach to knowledge acquisition. We acquire common information, standards, and practices directly from headquarters and then acquire the necessary information that is unique to individual airports.

These key issues present a novel and exciting problem in transitioning years of research to a highly complex domain [Jain *et al.*, 2010b; Yin *et al.*, 2010]. GUARDS is currently under evaluation by the TSA with the goal of incorporating its scheduling practices into their unpredictable security programs across airports nationwide.

2 Background

Game theory is a foundational approach used in multi-agent systems to reason about multiple agents each pursuing their own interests [Fudenberg and Tirole, 1991]. Game-theoretic approaches, specifically based on Stackelberg games, have recently become popular to address security problems (e.g. assigning checkpoints or air marshals) [Jain *et al.*, 2010b]. They model the commitment a security force must make in providing security and the attacker's capability of observing this commitment before attacking. The objective is to find the optimal mixed strategy to commit to given that an attacker will optimize his reward after observing this strategy. At this point we will describe how security games, as defined in [Yin *et al.*, 2010], fit into the Stackelberg paradigm.

In a security game there are two agents – the defender (security force) and an attacker – who act as the leader and the follower in a Stackelberg game. There is also a set of targets, which the defender is trying to protect. Each of these targets has a unique reward and penalty to both the defender and attacker, and the games are non-zero-sum. To protect these targets the defender has K resources at her disposal. There is a single security activity being considered and thus a target is either covered if a resource is used or uncovered otherwise by that activity. If the attacker attacks an uncovered target he gets his reward and the defender her penalty else vice versa. The defender's goal is to maximize her reward given that the attacker will attack with knowledge of the defensive strategy the defender has chosen. As in a regular Stackelberg game, the objective is to find the optimal mixed strategy to commit to given that an attacker will choose his optimal pure strategy.

There exist a number of algorithms and techniques for solving security games [Jain *et al.*, 2010a; 2010b; Conitzer and Sandholm, 2006]. DOBSS, a mixed-integer linear program is one of the general solvers and is capable of solving any Stackelberg game optimally [Jain *et al.*, 2010b]. Other algorithms are tailored to security games [Jain *et al.*, 2010b].

3 National Deployment Challenges

We now describe the three major issues in GUARDS; modeling, computational, and knowledge acquisition challenges.

3.1 Modeling TSA Resource Allocation Challenges

While we are motivated by an existing model of security games [Yin *et al.*, 2010], there are three critical aspects of the TSA domain that raise new challenges: (i) the defender now reasons over heterogeneous security activities for each potential area (target) within an airport; (ii) given the multiple possible security activities, the defender may allocate more than one resource per area; (iii) the defender now considers an adversary who can execute heterogeneous attacks on an area. For example, airports have ticketing areas, waiting areas, and cargo holding areas. Within each of these areas, TSA has a number of security activities to choose from such as perimeter patrols, screening cargo, screening employees and many others. The TSA must reason about a large number of potential threats in each area such as chemical weapons, active shooters, and bombs. The key challenge then is how to allocate limited TSA security resources to specific activities in particular areas, taking into account an attacker's response.

To address these challenges we create a more expressive model than outlined in security games. We refer to this new class of security games as Security Circumvention Games (SCGs). In SCGs, the defender must choose some combination of security activities to execute, where each security activity affects a specific area, and the attacker must reason over both which area to attack and which method of attack to execute based on the defender's strategy. At this time we elaborate on the defender's and attacker's possible strategies.

Defender Strategies: We denote the defender by Θ, and the set of defender's pure strategies by $\sigma_\Theta \in \Sigma_\Theta$. The TSA is able to execute a variety of security activities, which we denote by $S = \{s_1, \ldots, s_m\}$. Each security activity has two components. The first is the type of activity it represents, and the second is the area affected by the activity. We denote the set of areas by $A = \{a_1, \ldots, a_n\}$.

The defender has K resources available to run any K security activities which represents a single strategy $\sigma_\Theta \in \Sigma_\Theta$. For example, if there are three security activities, $S = \{s_1, s_2, s_3\}$ and two resources available, one possible pure strategy for the defender is to run s_1 and s_3. The TSA's task is to consider how to allocate these resources among security activities in order to provide the optimal protection to their areas. Similar to previous work, a mixed strategy (randomized solution) over Σ_Θ is typically the optimal strategy. Given that the number of possible combinations of K security activities at an airport can be on the order of 10^{13} or greater for the

TSA, we develop a compact representation of the possible strategies that we present in Section 3.2.

Attacker Actions: Defending a target against terrorist attacks is complicated by the diversity of the potential threats. For example, an attacker may try to use an active shooter, a suitcase bomb, or many others in any given area. We denote the attacker by Ψ, and the set of pure strategies for the attacker is given by $\sigma_\Psi \in \Sigma_\Psi$. Each pure strategy for the attacker corresponds to selecting a single area $a_i \in A$ to attack, and a specific mode of attack. However, given that each airport considers its own potential threats, enumerating all threats for each individual airport may not be practical. To handle the national deployment challenge we face and avoid this difficulty, we developed a novel way to represent threats for TSA's domain that we describe in Section 3.2.

3.2 Compact Representation for Efficiency

While we have developed a model that appropriately captures the TSA's security challenge, one issue with this model is that the defender strategy space grows combinatorially as the number of defender security activities increases. Also, on the attacker side, listing such a large number of potential threats would lead to extreme memory and runtime inefficiencies. Furthermore, existing solution techniques that have been developed for security games [Jain *et al.*, 2010a] are not directly applicable to Security Circumvention Games (SCGs). Hence we focus on a compact representation of SCGs for efficiency.

Threat Modeling for TSA: Given the large number of potential threats, the problem we face is how to model attack methods in a way that limits the number of threats GUARDS needs to reason over, but appropriately captures both an attacker's capabilities and his goals. We automatically generate attack methods for the adversary that capture two goals for the attacker: (i) an attacker wants to avoid the security activities that are in place; (ii) an attacker wants to cause maximal damage with minimum cost. To that end, the attacker's plan will be designed to avoid security activities that he believes will be in place. We will refer to this as circumventing security activities. For example, imagine there is a single area with three security activities such as passenger screening, luggage screening, and perimeter patrol. In this example, TSA only has one resource available and thus can only execute one of these activities at a time. While passenger screening may have the highest probability of success, if TSA never screens luggage or patrols the perimeter, the adversary can choose an attack path that avoids passenger screening such as utilizing a suitcase bomb or an attack from the perimeter.

Thus, the defender should avoid predictability, as attackers can exploit it. Hence, in GUARDS a list of threats are automatically generated in each area where each threat circumvents different combinations of specific security activities in that area. This avoids the issue of enumerating all the possible potential threats. However, we also incorporate a cost to the attacker for circumventing more activities to capture the idea of causing maximal damage at minimal cost. Each security activity has a circumvention cost associated with it and more activities circumvented leads to a higher circumvention cost. This cost reflects the additional difficultly of executing an attack against increased security due to factors like requiring additional resources, time and others for executing an attack. Since attackers can now actively circumvent specific security activities, randomization becomes a key factor because any deterministic strategies can be circumvented.

	$a_1:\emptyset$	$a_1:s_1$	$a_1:s_2$	$a_2:\emptyset$	$a_2:s_3$	$a_2:s_4$
s_1,s_2	2, -1	4, -3	4, -3	-20, 10	-17, 7	-17, 7
s_1,s_3	2, -1	-8, 3	4, -3	5, -5	-17, 7	8, -8
s_1,s_4	2, -1	-8, 3	4, -3	5, -5	8, -8	-17, 7
s_2,s_3	2, -1	4, -3	-8, 3	5, -5	-17, 7	8, -8
s_2,s_4	2, -1	4, -3	-8, 3	5, -5	8, -8	-17, 7
s_3,s_4	-10, 5	-8, 3	-8, 3	5, -5	8, -8	8, -8

Table 1: Example payoffs for sample game.

	$a_1:\emptyset$	$a_1:\gamma_1$	$a_2:\emptyset$	$a_2:\gamma_2$
γ_1,γ_1	2, -1	4, -3	-20, 10	-17, 7
γ_1,γ_2	2, -1	-2, 0	5, -5	-4.5, -5
γ_2,γ_2	-10, 5	-8, 3	5, -5	8, -8

Table 2: Example compact version of sample game.

Compact Representation: We introduce a compact representation that exploits similarities in defender security activities to reduce the number of strategies that must be considered when finding an optimal solution to SCGs. First, we identify security activities that provide coverage to the same areas, and have the same circumvention costs (i.e. have identical properties). Let $\Gamma = \{\gamma_1, \ldots, \gamma_t\}$ be the sets of security activities with identical properties. A strategy $\sigma_\Theta \in \Sigma_\Theta$ is represented by the number of resources assigned to each set of identical security activities $\gamma_i \in \Gamma$.

To illustrate this new representation, we provide a concrete example of the full representation versus the compact representation in Tables 1 and 2. In this example there are 4 security activities and 2 resources. Here, s_1 and s_2 have identical circumvention costs and affect a_1 while s_3 and s_4 have identical circumvention costs and affect a_2. Table 1 presents the full representation with corresponding payoffs and Table 2 represents the compact form of the same where γ_1 represents the group s_1 and s_2 and γ_2 represents the group s_3 and s_4. In both tables, each row represents a single pure strategy for the defender and each column the same for the attacker. Notice in Table 1 each strategy $\sigma_\Theta \in \Sigma_\Theta$ is represented by the exact security activities being executed while in Table 2 it is only which set $\gamma_i \in \Gamma$ each resource has been allocated to.

The key to the compact representation is that each of the security activities from a set $\gamma_i \in \Gamma$ will have the same effect on the payoffs. Therefore, it is optimal for the defender to distribute probability uniformly at random across all security activities within a set γ_i. Given that the defender strategy uniformly distributes resources among all security activities $s_j \in \gamma_i$ we also know that it does not matter which specific security activities the attacker chooses to circumvent from the set γ_i, only how many. For any given number of security activities circumvented, the expected payoff to the attacker is identical regardless of which specific activities within the set are chosen. Thus, we can use a similar compact representation for the attacker strategy space, reasoning only over the

aggregate number of security activities of each type circumvented rather than specific security activities circumvented.

Given this, we only need to know how many security activities are selected from each set in order to compute the expected payoffs for each player in the compact representation. For example, examining the second row and second column of Table 2 we see that the reward to the defender is -2 and the reward to the attacker is 0. In this case, the defender strategy is to assign 1 resource to activities in γ_1 and 1 resource to activities in γ_2. Given that she is uniformly distributing these resources, it follows that she will execute s_1 half of the time and s_2 the other half. On the attacker side, we know that the attacker is circumventing one security activity from the set γ_1. If he circumvents either s_1 or s_2 he will only succeed half of the time. Thus, half of the time the defender receives 4 and the other half -8 for an expectation of -2 $(4 * .5 + (-8) * .5)$. We compute the attacker's reward in the same manner.

3.3 Knowledge Acquisition

One of the most difficult issues we faced from a potential national deployment perspective was in acquiring the appropriate knowledge for the security challenge being considered. Unfortunately, with hundreds of airports to consider, it is not possible to sit down at each location and acquire the exact needs for each of them. To overcome this obstacle, in close collaboration with TSA headquarters we developed a two phase knowledge acquisition process.

In phase one, we take an approach similar to previous centralized approaches [Jain *et al.*, 2010b]. In particular, we met with domain experts to acquire knowledge that is common among all airports. This included area definitions, defining security activities, and determining resource capabilities among others. In phase two of our knowledge acquisition, we took a decentralized approach where it is the responsibility of individual airports to input customized information. We rely on SCGs and developed a system in collaboration with headquarters that allows individual airports to manipulate specific components within this framework to create unique game instances. These inputs are designed to ensure that individual airports maintain standards set forth by headquarters in phase one. For example, individual airports are responsible for determining the unique reward and penalty associated with each area for the defender and attacker given a successful or unsuccessful attack. This is achieved by answering questions related to the number of fatalities that may result, whether the area has access control, and others.

Our two phase knowledge acquisition process follows a partially centralized approach and provides the following advantages: (i) it allows domain experts from TSA headquarters to assure that the system meets the required needs of the challenge being considered; (ii) it focuses on creating customizable inputs instead of a system tailored to a highly specific problem instance; (iii) it allows TSA headquarters control while still enabling individual airports to customize the system to meet the airports individual needs.

4 System Architecture

The GUARDS system consists of three modules:

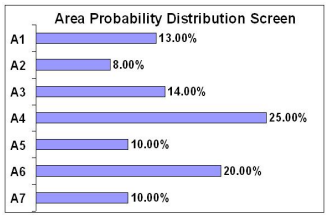
Input Module: The input module is composed of three classes of inputs that are required by the system: (i) area data; (ii) security activity data; (iii) resource data.

Back-end Module: The back-end module has three components: (i) generating the game; (ii) solving the game; (iii) returning a sample schedule to TSA. GUARDS uses DOBSS [Jain *et al.*, 2010b] to solve a generated game instance.

Display/Output Module: The actual resource assignment selected is presented to the user via the display/output module. The generated schedule is displayed two ways. First as a summary of the number of resources assigned to each area similar to the mockup in Figure 1 (a)[1]. Second as an in depth report of the schedule including the specific security activities that were chosen. TSA personnel can also choose to examine the distribution of resources over areas that the optimal mixed strategy provides as in Figure 1 (b).

Locations	8/15	8/16	8/17	8/18	8/19	8/20	8/21
A1	1	3	1	1	3	0	0
A2	0	2	1	0	0	0	0
A3	2	2	0	1	0	3	1
A4	1	0	3	2	0	1	2
...	...	...	...	...	...	...	...
Total	10	10	10	10	10	10	10

(a) Summary of sample schedule



(b) Summary of area probability

Figure 1: Example display

5 Evaluation

When evaluating a system like GUARDS there are two important issues that are raised: (i) scalability and run-times; (ii) evaluating the value of the security policies generated against alternative approaches.

5.1 Run-time Analysis

We present simulation results focusing on the computational efficiency of our compact method versus the full representation. All experiments are run on a system with an Intel 2 GHz processor and 1 GB of RAM. We used a publicly available linear programming package called GLPK to solve optimization problems as specified in the original DOBSS procedure. For the compact version we use a slightly altered version of DOBSS that is designed specifically for efficiency in the compact representation. The solver was allowed to use up to 700 MB of memory during the solution process. For larger game instances the full representation runs out of memory and so we exclude results for these cases. In all experiments both the solution found by the full representation and the solution found by the compact representation are optimal.

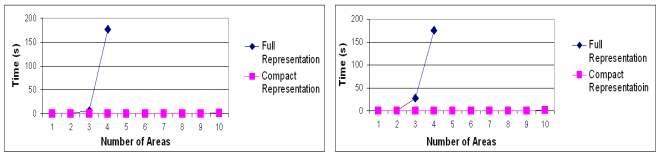
To test the solution methods we generated and averaged 20 random game instances by randomly selecting payoff values from 1 to 50 and circumvention costs from 1 to 5 for each area in each instance. We considered three different scenarios: (i) we increase the number of areas, where each area has exactly 3 security activities with identical properties and there are 5 available resources; (ii) identical to (i), except that security activities are distributed randomly across possible areas,

[1]We are unable to show real screen shots for security reasons.

holding fixed the total number of security activities as three times the number of areas; (iii) a situation with 10 areas to protect, each area has 3 identical security activities, and we increase the number of resources available to distribute between these areas. Given these three scenarios there can be upwards of 142,506 defender pure strategies and 70 attacker pure strategies in the full representation.

Examining Figure 2 (a), we show the improvement in runtime of our compact representation over the full representation in our first scenario. Similarly, in Figure 2 (b), we see analogous benefits for the compact representation in our second scenario. Finally, in Figure 3, we show our third scenario where we also see improvement in run-time for our compact representation over the full representation.

These results show the benefits of our compact representation in terms of efficiency. We obtained further efficiency gains by caching results: specifically, the inputs into the game do not change on a daily basis. Thus, we can cache the resulting mixed strategy, and present results from sampling this strategy, as long as users have not changed the inputs.



(a) Three activities per area (b) Random activities per area

Figure 2: X-axis: Areas, Y-axis: Run-time in seconds

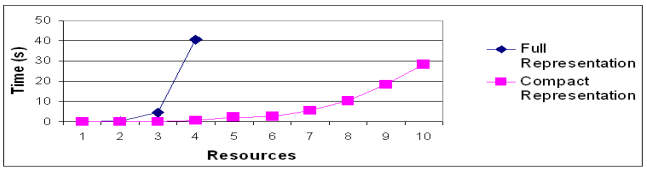


Figure 3: X-axis: Resources, Y-axis: Run-time in seconds

5.2 Security Policy Analysis

For this analysis we examined the security policies generated by our game representation against two other possible solution strategies. The first strategy is a solution concept where resources are distributed uniformly among areas (uniformly random), an approach sometimes used in lieu of a game-theoretic approach. The second strategy uses our new representation, however, it does not allow attackers to circumvent security activities (SCGs without circumvention). This is a simplified model of an attacker where the attacker does not plan around specific security measures. Finally, we included our new representation of SCGs.

We generated 20 random game instances with 10 areas and 3 security activities per area. The payoff value of each area for both the defender and attacker are randomly selected from 1 to 50 and the circumvention costs are similarly selected from 1 to 5. We then calculated the optimal solution under the current solution strategy (i.e. uniformly random, SCGs without circumvention, and SCGs). After finding the optimal solution, we determined the expected reward for each solution given the assumptions made in SCGs (i.e. an attacker is allowed to circumvent specific security activities when planning his attack). For each game instance, we computed solutions for varying number of resources from 1 to 10 as seen on the x-axis of Figure 4. On the y-axis, we present the average expected reward obtained by each solution strategy across all 20 game instances. In Figure 4 we see that SCGs highly outperform the other two models obtaining an over 200% improvement in reward with 10 resources.

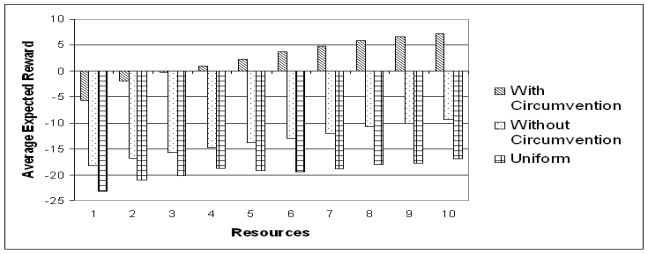


Figure 4: Policy Analysis: Increasing resources for 10 areas

6 Lessons in Turning Research into Practice

GUARDS is the result of a unique collaboration where university researchers worked directly with a security agency for the purpose of creating a useful product to potentially deploy outcomes of research on a national scale. This collaboration to transition research to such a large-scale deployment has presented valuable lessons. This section outlines the three areas of insights we have gained in the process: (i) acceptance of GUARDS at headquarters; (ii) acceptance of GUARDS by a variety of end-users at numerous airports; (iii) obtaining correct input from users. Some of these insights are contrary to accepted wisdom in the research community.

In a large organization like the TSA that deals with important security issues, quality guarantees are important. Furthermore, optimality is highly desirable or even a design requirement. This is in contrast to a common assumption by researchers that speedy heuristic solutions that are on average high quality may be adequate "in the field". Without guarantees, the TSA may be unable to justify the use of a security strategy. Thus, we use a solver known as DOBSS, which provides game-theoretic optimal solutions in Stackelberg games.

With respect to acceptance of GUARDS at individual airports, one major lesson learned is bridging the culture gap in academic research and real-world operations. Indeed, what researchers may consider small uninteresting issues may nullify all their major research advances. For example, in an initial version of GUARDS, we displayed the final probabilities of our mixed strategies, but truncated the presentation of real numbers (i.e. truncating all decimal values). The caused users to assume GUARDS was not utilizing all resources and thus was incorrect. A second major lesson learned is the continued need for efficiency of game-theoretic algorithms. While significant research has gone into speeding up these algorithms, there still does not exist off-the shelf algorithms; GUARDS required the use of new compact representations. We have outlined our key advances in this regard in Section 5.1; including the need for caching.

A third lesson learned in user acceptance is careful design

of the user interface so as to reduce the amount of user workload to provide inputs. For instance, if users are required to directly enter values into the generated game matrix it can require thousands of inputs. Instead, it is important to provide a user-friendly method of conveying the necessary information. We used a simple interface where users are only required to input the base information (e.g. areas and security activities) that is then used to generate the larger game matrix. This is information that users have direct access to and can easily be input by the individual airports.

Finally, in any collaboration, it is important that researchers are able to obtain the appropriate input from their collaborators. This includes understanding what information is available versus what is not and accounting for this in modeling of the problem. Also, often end-users will not understand the techniques being applied and thus are prone to providing vague or incorrect information. For example, given a 10 point utility scale, when asking a security agency such as the TSA to provide a utility for an attacker and themselves as a defender on a successful attack, they may always claim that it is -10 for them and 10 for the attacker on every area. In practice, this feedback may not be useful because attacks on different areas may actually have very different impact in terms of economic damage, casualties, and many other factors. To aid in preventing this scenario, it is important to convey the impact that inputs will have on outputs; aiding their understanding of how their inputs will affect the results.

7 Related Work and Summary

TSA is charged with protecting over 400 airports in the US. The key challenge is how to intelligently deploy limited security resources to security activities in a risk-based manner, yet maintain unpredictability and provide optimal protection. These decisions may be made on a daily basis, based on the local information available at each airport. We present an application, GUARDS, that represents promising potential for transitioning years of academic research into an application designed for potential national scale deployment. Our work complements previous research and applications that utilize a game-theoretic framework. Specifically, much work has been done exploring traditional security games [Yin *et al.*, 2010; Korzhyk *et al.*, 2010] and utilizing these models to deploy applications for standalone locations such as ARMOR and IRIS [Jain *et al.*, 2010b]. Our work also complements research actually applied to randomize patrolling strategies in robot patrol [Agmon *et al.*, 2009; Basilico *et al.*, 2009], given our emphasis on modeling adversaries in a game-theoretic setting.

In creating GUARDS, we address three key issues that arise from a potential national deployment case. These issues are: (i) appropriately modeling TSA's security challenge to achieve the best security policies; (ii) efficiently finding solutions to the problem we consider; (iii) knowledge acquisition for hundreds of end-users under one organization. To address these challenges we develop a novel game-theoretic model, referred to as Security Circumvention Games (SCGs), design an efficient solution technique for reasoning over our new game model based on creating a compact representation of the game, and utilize a two phase knowledge acquisition process for acquiring appropriate domain knowledge. To conclude, we present results demonstrating the benefits of our contributions along with lessons learned in creating GUARDS. The scheduling assistant has been delivered to the TSA and is currently under evaluation and testing for unpredictable scheduling practices at an undisclosed airport.

8 Acknowledgements

This research was supported by the United States Department of Homeland Security through the National Center for Risk and Economic Analysis of Terrorism Events (CREATE) under award number 2010-ST-061-RE0001. However, any opinions, findings, and conclusions or recommendations in this document are those of the authors and do not necessarily reflect views of the United States Department of Homeland Security, the University of Southern California, or CREATE.

References

[Agmon *et al.*, 2009] Noa Agmon, Sarit Kraus, Gal Kaminka, and Vladimir Sadov. Adversarial uncertainty in multi-robot patrol. In *IJCAI*, 2009.

[AIR, 2011] Air traffic control: By the numbers. In *http://www.natca.org/mediacenter/bythenumbers.msp#1*, 2011.

[Basilico *et al.*, 2009] Nicola Basilico, Nicola Gatti, Thomas Rossi, Sofia Ceppi, and Francesco Amigoni. Extending algorithms for mobile robot patrolling in the presence of adversaries to more realistic settings. In *IAT*, 2009.

[Conitzer and Sandholm, 2006] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *EC*, 2006.

[Fudenberg and Tirole, 1991] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991.

[Jain *et al.*, 2010a] Manish Jain, Erim Kardes, Christopher Kiekintveld, Milind Tambe, and Fernando Ordóñez. Security games with arbitrary schedules: A branch and price approach. In *AAAI*, 2010.

[Jain *et al.*, 2010b] Manish Jain, Jason Tsai, James Pita, Christopher Kiekintveld, Shyamsunder Rathi, Fenando Ordóñez, and Milind Tambe. Software assistants for randomized patrol planning for the LAX airport police and the Federal Air Marshals Service. volume 40, pages 267–290, 2010.

[Korzhyk *et al.*, 2010] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *AAAI*, 2010.

[TSA, 2011] TSA — Transportation Security Administration — U.S. Department of Homeland Security. In *http://www.tsa.gov/*, 2011.

[Yin *et al.*, 2010] Zhengyu Yin, Dmytro Korzhyk, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness. In *AAMAS*, 2010.